



PM Challenge 2009

Managing Software

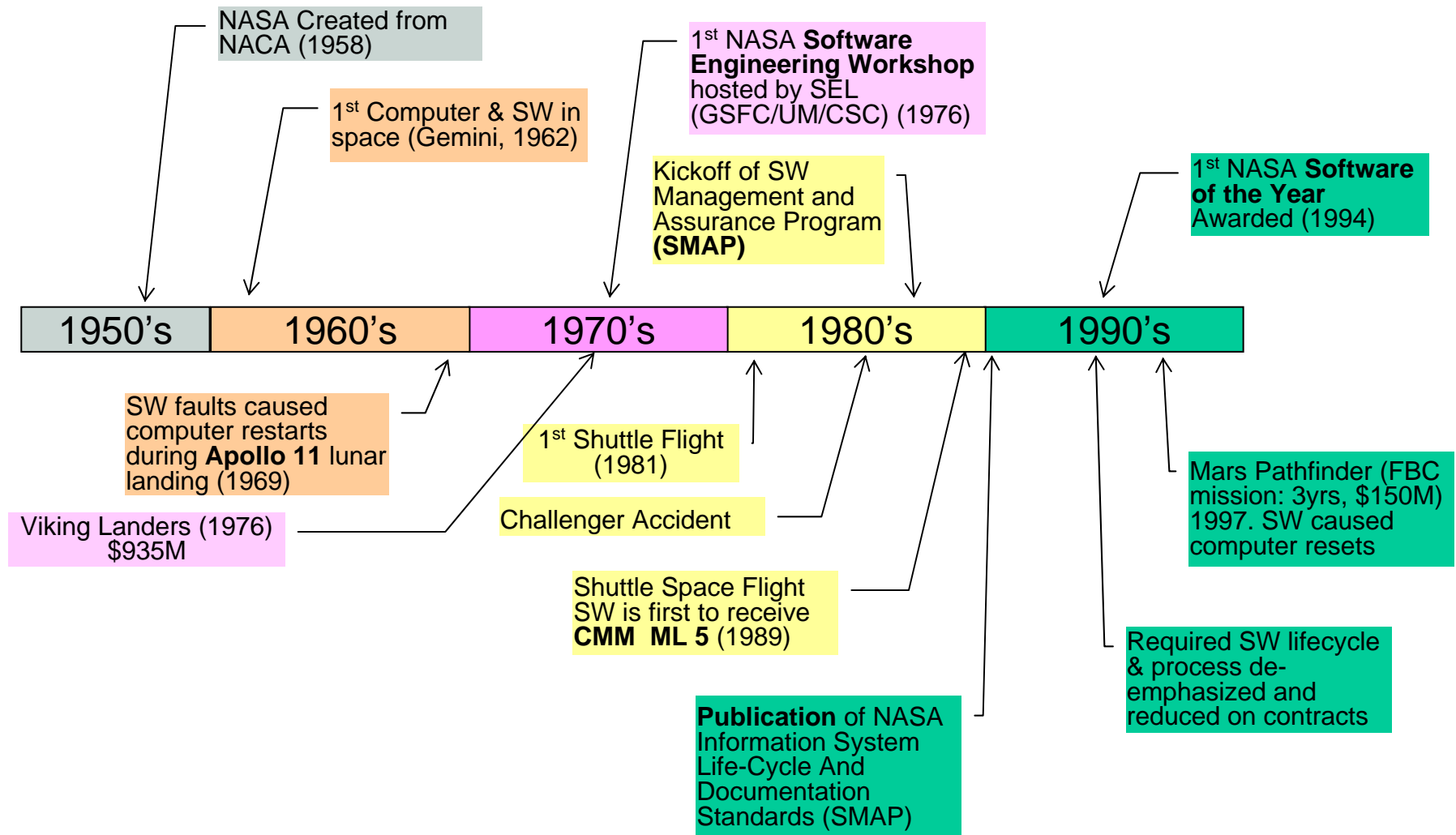
***assessing the software
components on your project***

February 25, 2009

John C. Kelly & Tim Crumbley
Office of Chief Engineer

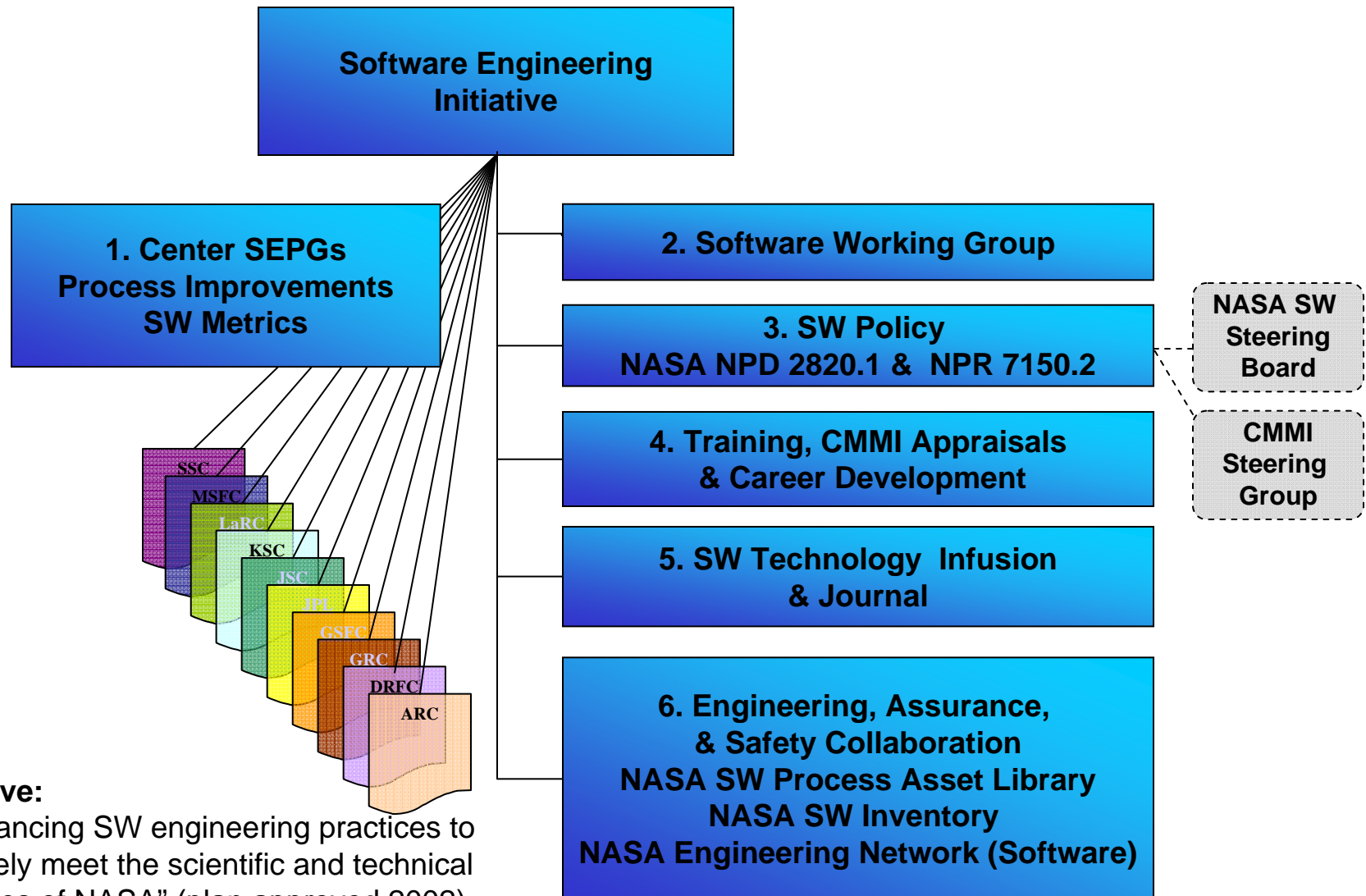


Timeline 1957 – 1998 (Relevant to NASA Software Engineering)





Software Engineering Initiative



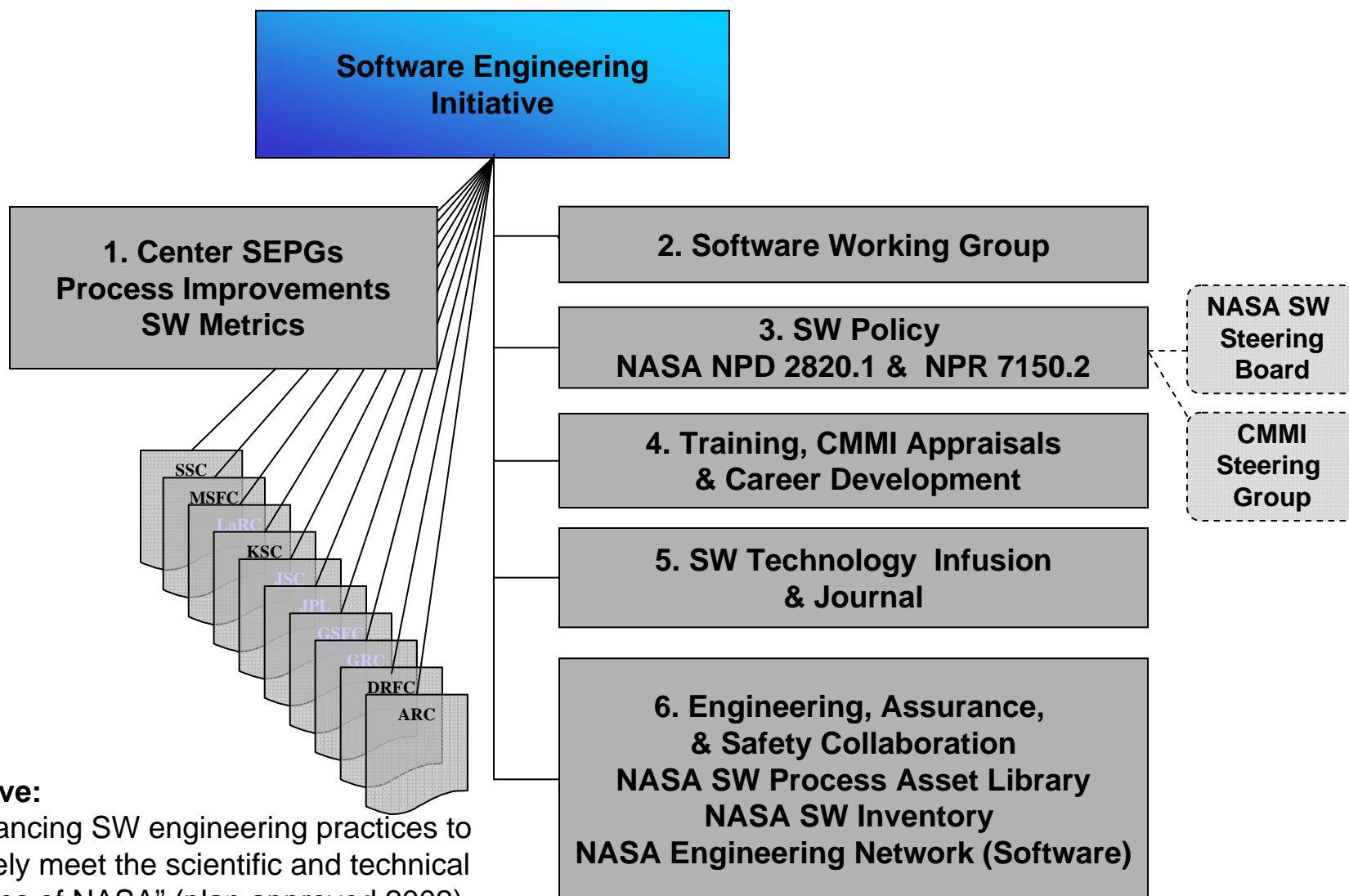
Objective:

"...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer



Software Engineering Initiative



Objective:

"...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer



Goal of the NASA Software Engineering Initiative

- **Advance software engineering practices (development, assurance, and management) to effectively meet the scientific and technological objectives of NASA**
 - Consistent performance for software products engineered for or by NASA in the areas of:
 - **Schedule**
 - **Cost**
 - **Delivered Functionality**
 - **Quality**
 - Infuse improved technology
 - Agency use of best practices for Software Engineering
 - Skilled and knowledgeable workforce



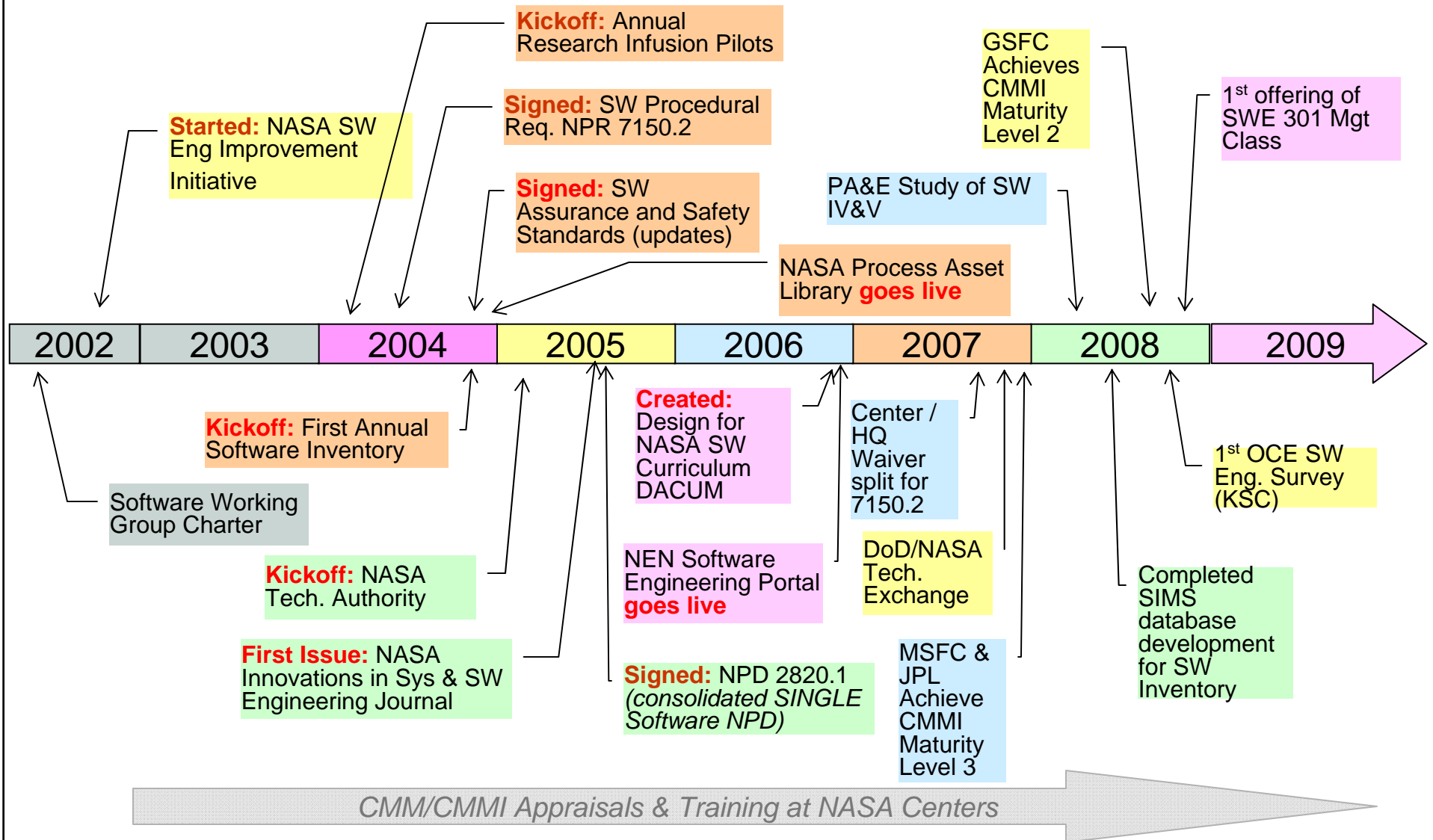
Key Impacts

1. **Projects are engaged in using the improvements that the Software Engineering Initiative has developed**
2. **Software development planning has been improved across the Agency**
 - “Vast improvement in planning of software projects and in monitoring progress” – GSFC
3. **NASA’s contractor community has heard the word that the bar has been raised with respect to software engineering and is responding appropriately**
 - Solicitations include statements flowing down consistent software requirements
4. **A solid foundation and structure is now in place for developing software in a disciplined manner**
 - NASA PAL Website with more than 650+ process assets (including checklists, templates, process documentation, guidance, etc.)
5. **The Agency is better prepared for major programs and projects than it was before NPR 7150.2, Software Engineering Requirements**
6. **The knowledge and skills of the NASA software engineering community has significantly improved**



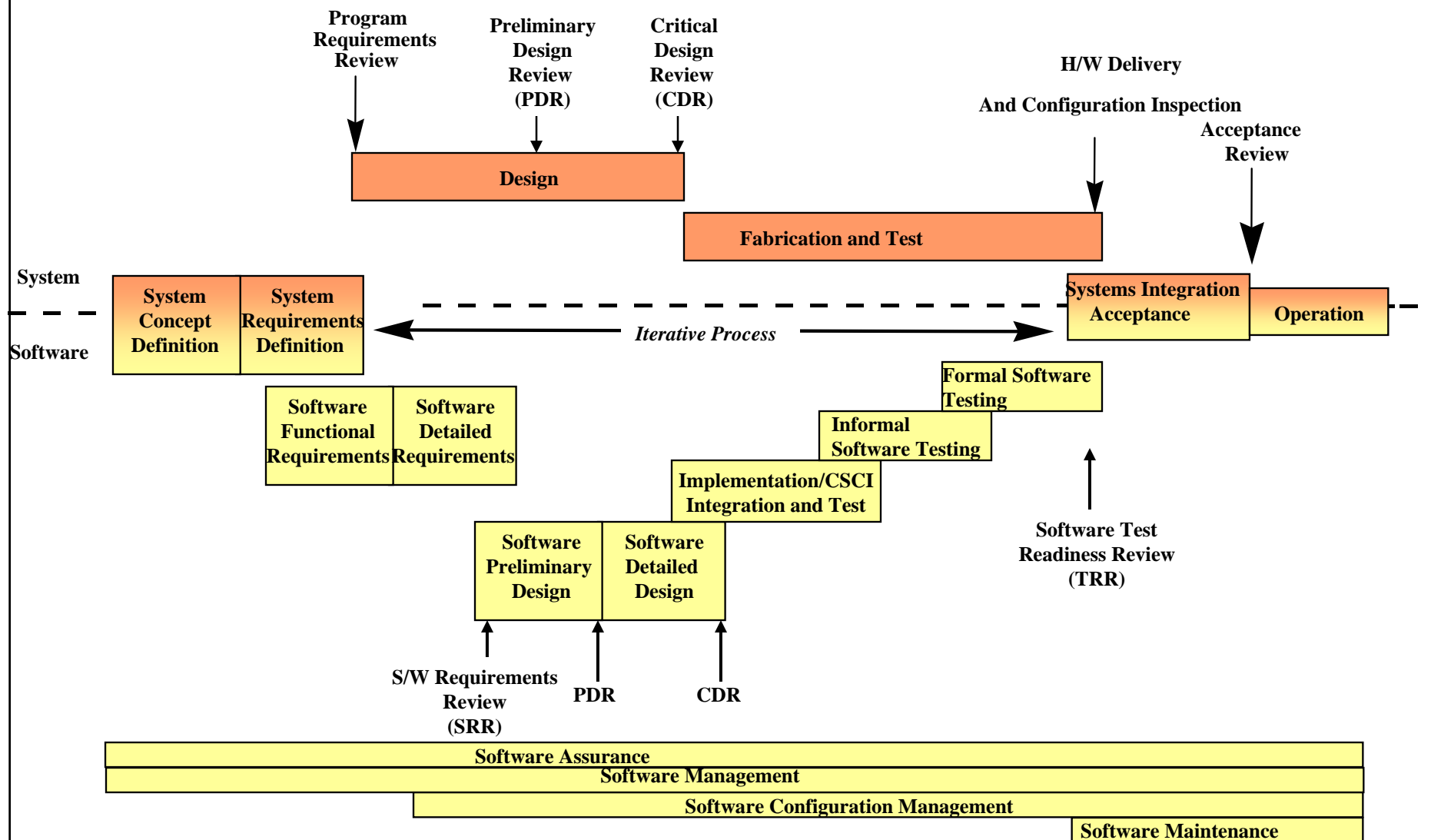
Timeline 2002 – 2008

NASA Software Engineering Initiative



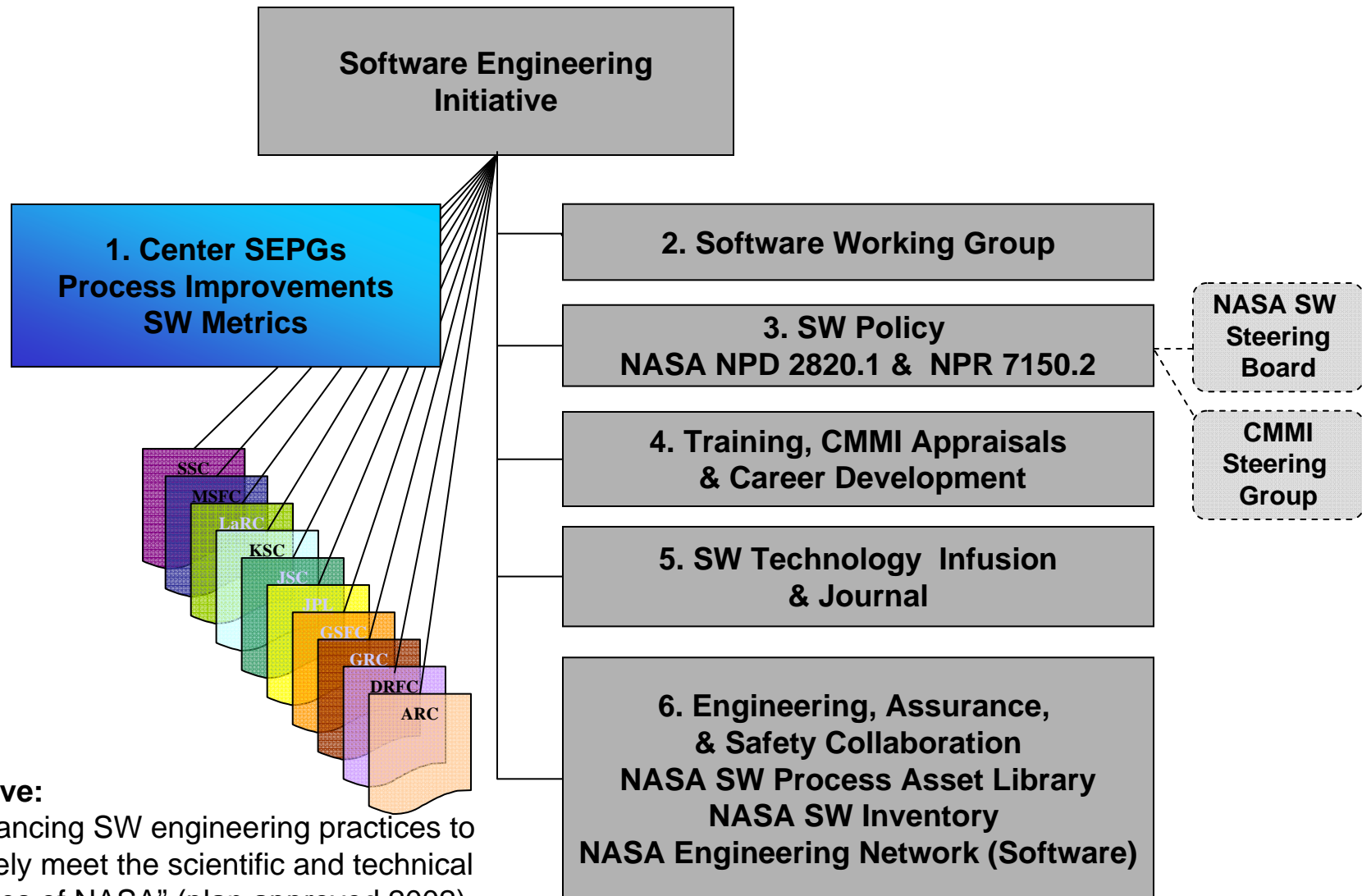


Software Development Life Cycle





Software Engineering Initiative



Objective:

"...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer



GODDARD
SPACE FLIGHT CENTER



GSFC Software Process Improvement

If the Process Works...Improve It.



+ GSFC SW
IMPROVEMENT

+ PROCESS
ASSETS LIBRARY

+ TRAINING

TOOLS

+ MEASURES

+ LESSONS
LEARNED

SPI Initiative

The Navigation Bar

Site

+ S

+ F

Co

Training to ensure
project engineers
have necessary skills

News and Training Events

Update

"Brand

"Fund

"Physi

"Baseline Audit Checklist"

Local tools to
support projects

[Link to New Assets --->](#)

Welcome

The goal of GSFC Software Process Improvement is to establish and continuously improve system and software processes and products by providing the necessary supporting infrastructure, such as tools, templates, measurements support, and lessons learned. The objectives are to:

- Improve the quality, reliability, and safety of our products through the integration of sound system and software engineering principles and standards, so that our customers receive highly effective and reliable products that fulfill their scientific and

Conferences

Carnegie Mellon Software Engineering Institute (SEI) Software Engineering Process Group North America conference (SEPG '09)

March 23-26, 2009
San Jose, California, USA

The 21st Annual Systems & Software Technology Conference (STC 2009)

April 20-23, 2009
Salt Lake City, Utah, USA

Southeastern Software & Systems Engineering



Requirements Flow

**NASA
SWG**

External
Input

NASA Policies & Directives

Industry Standards

- ISO 9001:2000, Quality Management Systems
- IEEE / EIA 12207, Standard for Info. Tech
- Capability Maturity Model® Integration (CMMI)

Institutional
Directives

Flight Project Practices
(Applicable to Software)

Principles for Flight Systems
(Applicable to Software)

Software Development Requirements (SDR)

**Center
SEPG**

Institutional
Assets

JPL Standard Software Process

Guidance Material

- Templates
- Handbooks / Guides
- Compliance Matrices
- Sample Documents
- Example Text

Project & Line
Directives

Domain (Section / Group)
Specific Requirements /
Procedures

Project Specific
Requirements / Procedures

**Project
Team**

Project or Task
Software Plans

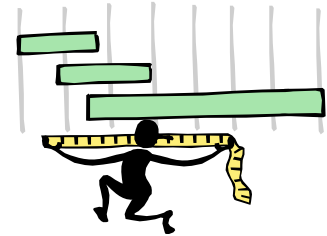
Supporting Plans

Supporting Plans

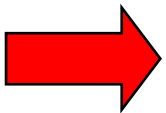
SMP



What is this CMMI?

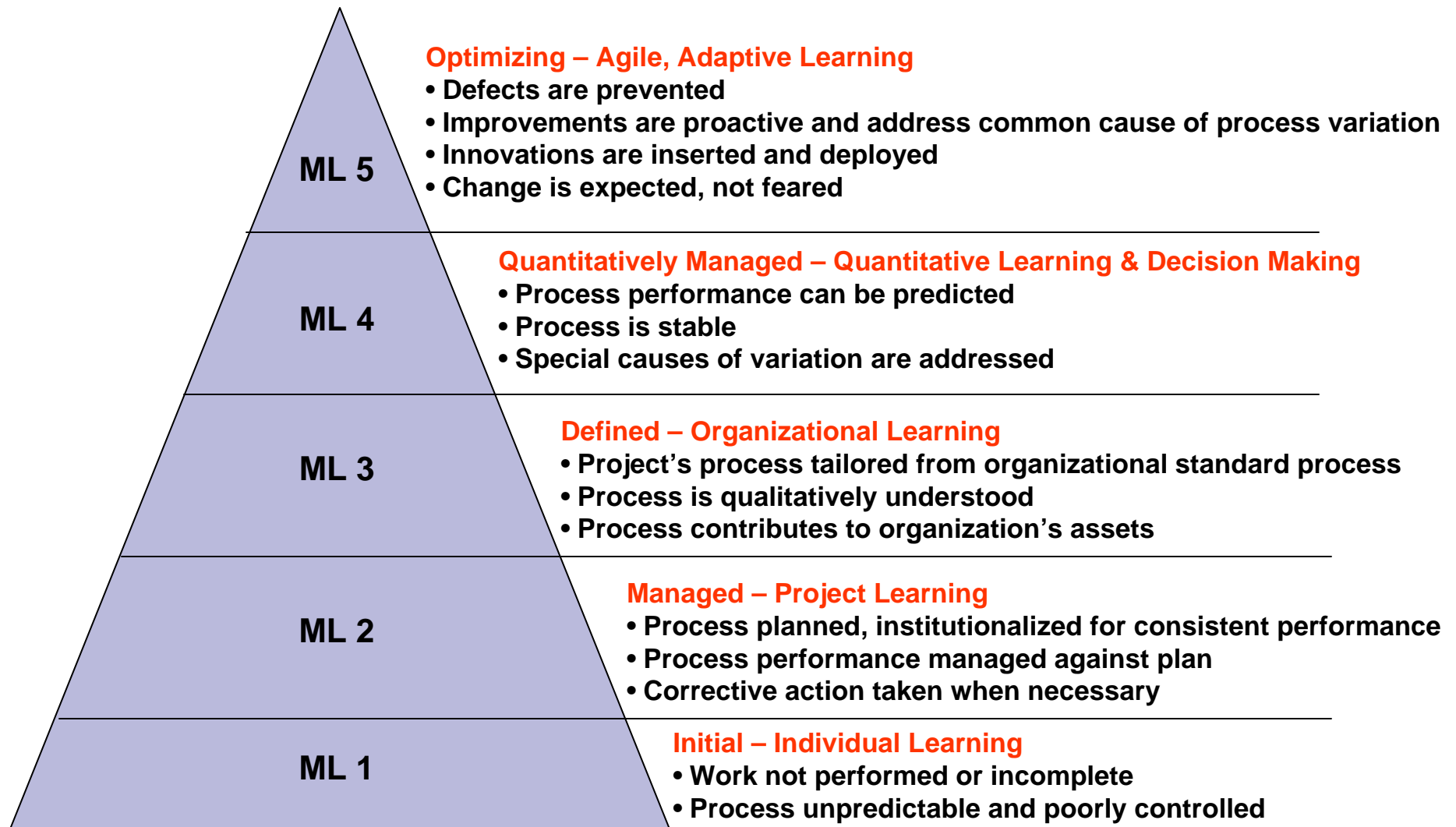


- **What it is:** A process improvement model that provides a set of industry-recognized practices to address productivity, performance, costs and stakeholder satisfaction in the systems engineering and software development process.
 - Helps your organization examine the effectiveness of your processes
 - Establishes priorities for improvement
 - Helps you implement these improvements
- **How it is different:** The CMMI provides an integrated, consistent, enduring framework for enterprise-wide process improvement and can accommodate new initiatives as future needs are identified. Unlike single-discipline or stove-pipe models that can result in confusion and higher costs when implemented together.
- **Who it is for:** Those providing systems and software engineering products and services to organizations who transform customer needs expectations, and constraints into products, and supporting these products throughout their life. **If you manufacture, code, analyze, maintain or document a product, you need this!**





CMMI Maturity Level Characterizations





NASA CMMI Summary

as noted on SEI website for Appraisals Disclosure Statement

Center	Rating	App. Date	# Projects	Type	Project sizes (# people)
MSFC	ML3	24-Aug-07	3	Development	57, 4, 2 = 63
GSFC	ML2 + RSKM(2)	16-May-08	4	Dev & Maintenance	25, 2, 18, 8 = 53
JPL	ML3	27-Sep-07	7	Dev & Maintenance	1, 8, 11, 17, 16, 1, 30 = 84
LaRC - FSSB	ML2 + CL3	3-Oct-08	3	Services	1, 1, 3 = 5
LaRC - ASDC	PP(CL3), CM(CL1)	17-Nov-06	1	Data Center Support	85
LaRC - FSSB (renamed SDAB)	REQM(CL3), CM(CL3)	17-Mar-06	4	not given, older format	not given, older format

Planned CMMI Appraisals in FY09

Title	Type	days	start date	Center
SCAMPI A	appraisal	5	6/2009	MSFC
SCAMPI A	appraisal	7	3/2009	JSC
SCAMPI A	appraisal	8	9/2009	KSC
SCAMPI B	appraisal	4	3/2009	JPL
SCAMPI B Pre-Appraisal	pre- appraisal		4Q FY09	ARC
SCAMPI C	appraisal	5	9/2009	JPL
SCAMPI C Pre-Appraisal	pre- appraisal		2Q FY09	ARC



Example Requirement from NPR 7150.2 (Metrics)

“5.3.1 Software Metrics Report

The Software Metrics Report provides data to the project for the assessment of software cost, technical, and schedule progress. The Software Metrics Report shall contain as a minimum the following information tracked on a CSCI basis: [SWE-117]

- a. Software progress tracking measures.
- b. Software functionality measures.
- c. **Software quality measures.**
- d. Software requirement volatility.
- e. Software product characteristics.”*

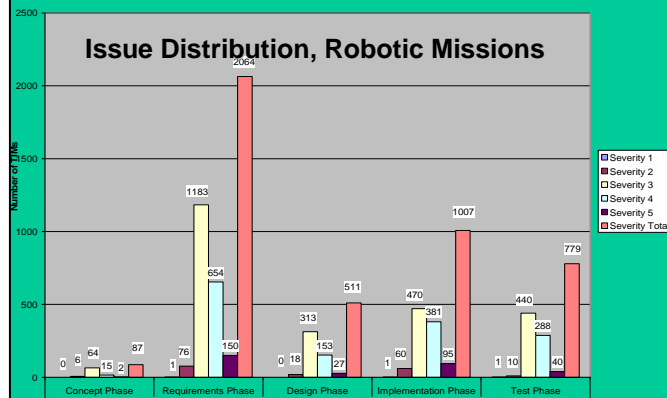
Note: “An example set of software quality measures that meet 5.3.1.c include, but are not limited to:

- a. Number of software Problem Reports/Change Requests (new, open, closed, severity).
- b. Review of item discrepancies (open, closed, and withdrawn).
- c. Number of peer reviews/software inspections (planned vs. actual).
- d. Peer review information (e.g., effort, review rate, defect data).
- e. Number of software audits (planned vs. actual).
- f. Software audit findings information (e.g., number and classification of findings).
- g. Software risks and mitigations.
- h. Number of requirements verified or status of requirements validation.”

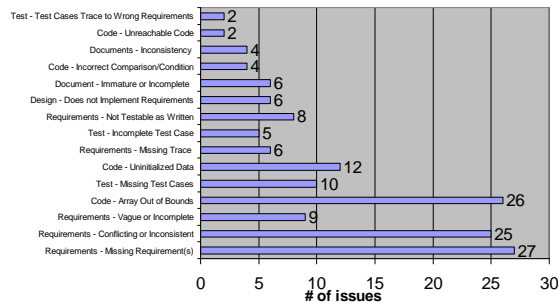
**Required only for Class A, B, F (not OTS), & G (not OTS) software*

Metrics Collection

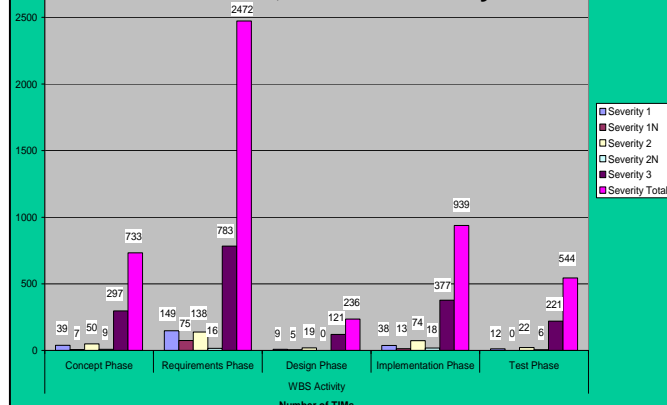
Agency Wide Metrics (IV&V)



Robotic Mission Issue Types – Severity 1 and 2

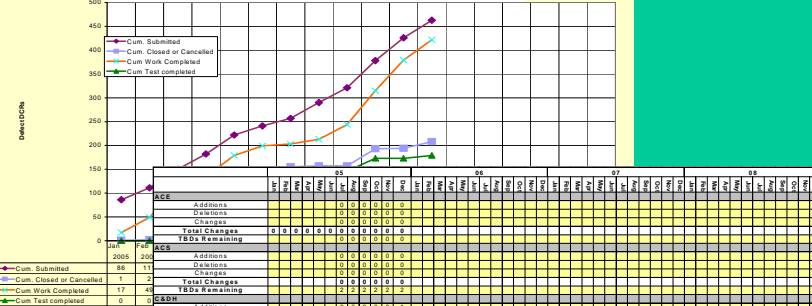


Issue Distribution, Human-rated Systems



Center/Project Metrics

Overall Defects Correction Trend



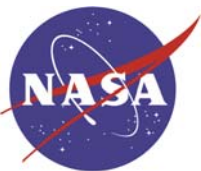
Requirements Volatility Data

Constellation Program

399 **Management performance and product measures.** Each project shall report
 400 status using the program's computing system management performance and
 401 product quality measures in Appendix E. [CSR-32-003]

APPENDIX E COMPUTING SYSTEM DEVELOPMENT METRICS

1142 This appendix addresses the measures that are of interest to the Constellation
 1143 program as indicators primarily of product quality and management performance,
 1144 and are designed to be consistent with the requirements set forth in this
 1145 document.



Metrics Implementation on Projects

Ares I Software Management Plan (SMP)

CxP 72051

BASELINE (DRAFT)

APPENDIX G SOFTWARE METRICS DRD

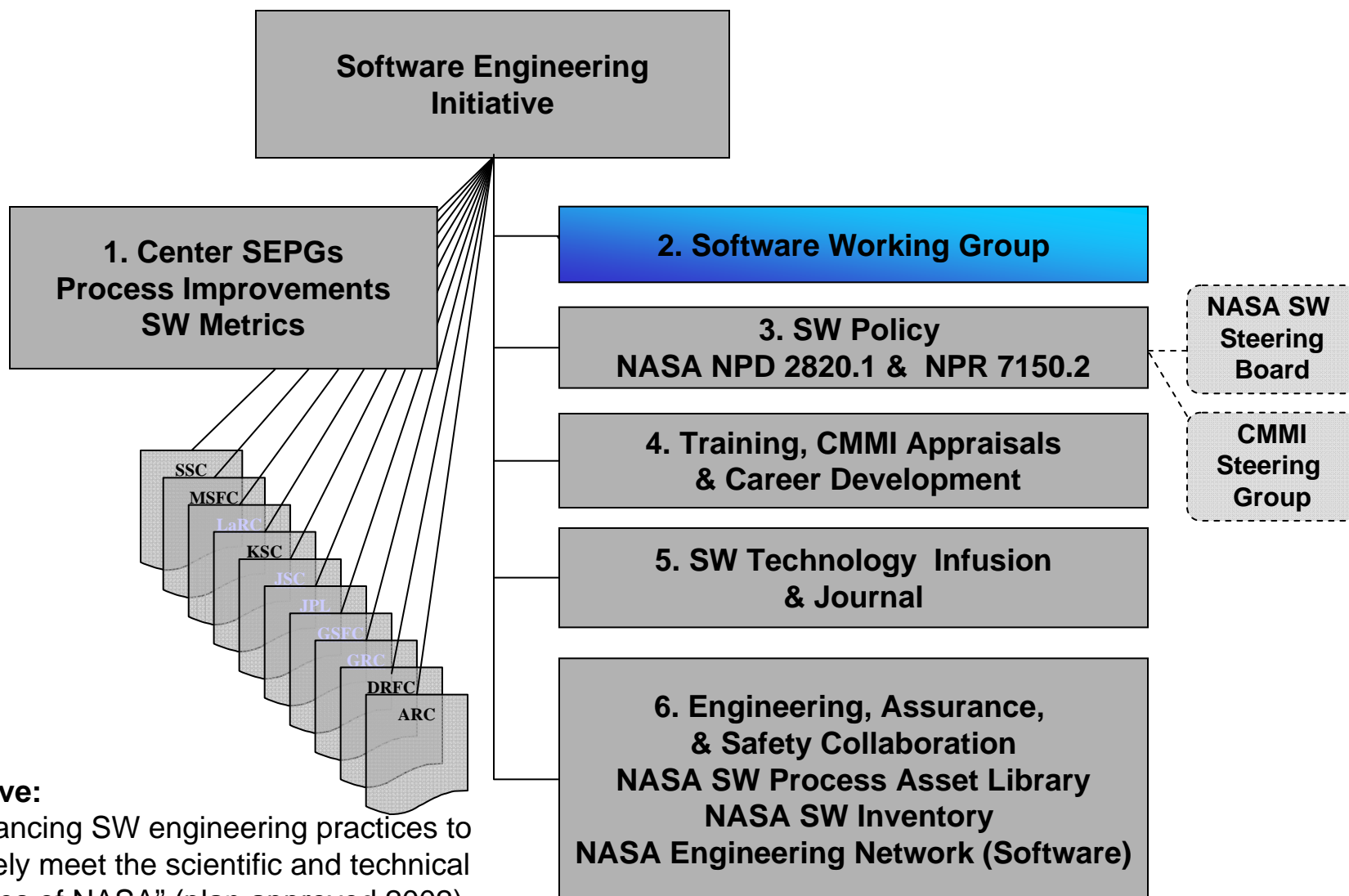
RELEASE DATE: TBD

The Software Metrics Report DRD requires the following information to be tracked on each CSCI (reference STD/SW-SMR):

- a. Software resources – planned vs. actual.
- b. Requirements stability – number of software requirements, number of requirements changes, and number of TBDs.
- c. Software size in agreed upon measurement units (e.g. source lines of code (SLOC)) – planned vs. actual.
- d. Computer resource utilization in percentage of capacity (e.g. memory, processor, I/O throughput) with agreed upon thresholds.
- e. Number of test cases successfully developed, dry run, and used for formal test – planned vs. actual.
- f. Number of Computer Software Units (CSUs) in design phase, in coding phase, coded and under Configuration Management (CM), completed Unit Test, and Integrated into Computer Software Configuration Item (CSCI).
- g. Number of software problem reports and review item discrepancies – open, closed and withdrawn.
- h. Number of Peer Reviews/Software Inspections – planned vs. actual.
- i. Number and classification of findings found during Peer Reviews/Software Inspections.
- j. Number of Software Audits – planned vs. actual.
- k. Number and classification of findings found during Software Audits.
- l. Software Risks and mitigations
- m. Software development schedule tasks – planned vs. actual.
- n. Other information that may assist the government in evaluating the contractor's cost, technical and schedule performance, e.g., innovative processes and cost reduction initiatives.
- o. System performance indicators, e.g., subsystem to subsystem data latency metrics.



Software Engineering Initiative



Objective:

"...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer



Roles of the Software Working Group

- Function as an advisory group
- Recommend, draft as requested, review, and promote policies, standards, & best practices
- Recommend and provide technical support for special studies
- Focus, integrate, and promote innovation and the continuous improvement of NASA's software engineering processes
- Support and help guide the establishment of software process improvement programs at each Center
- Facilitate the transfer of software technology
- Coordinate NASA representation within agency, interagency, and international boards
- Provide information to improve communication on software issues
- Ad hoc activities as needed

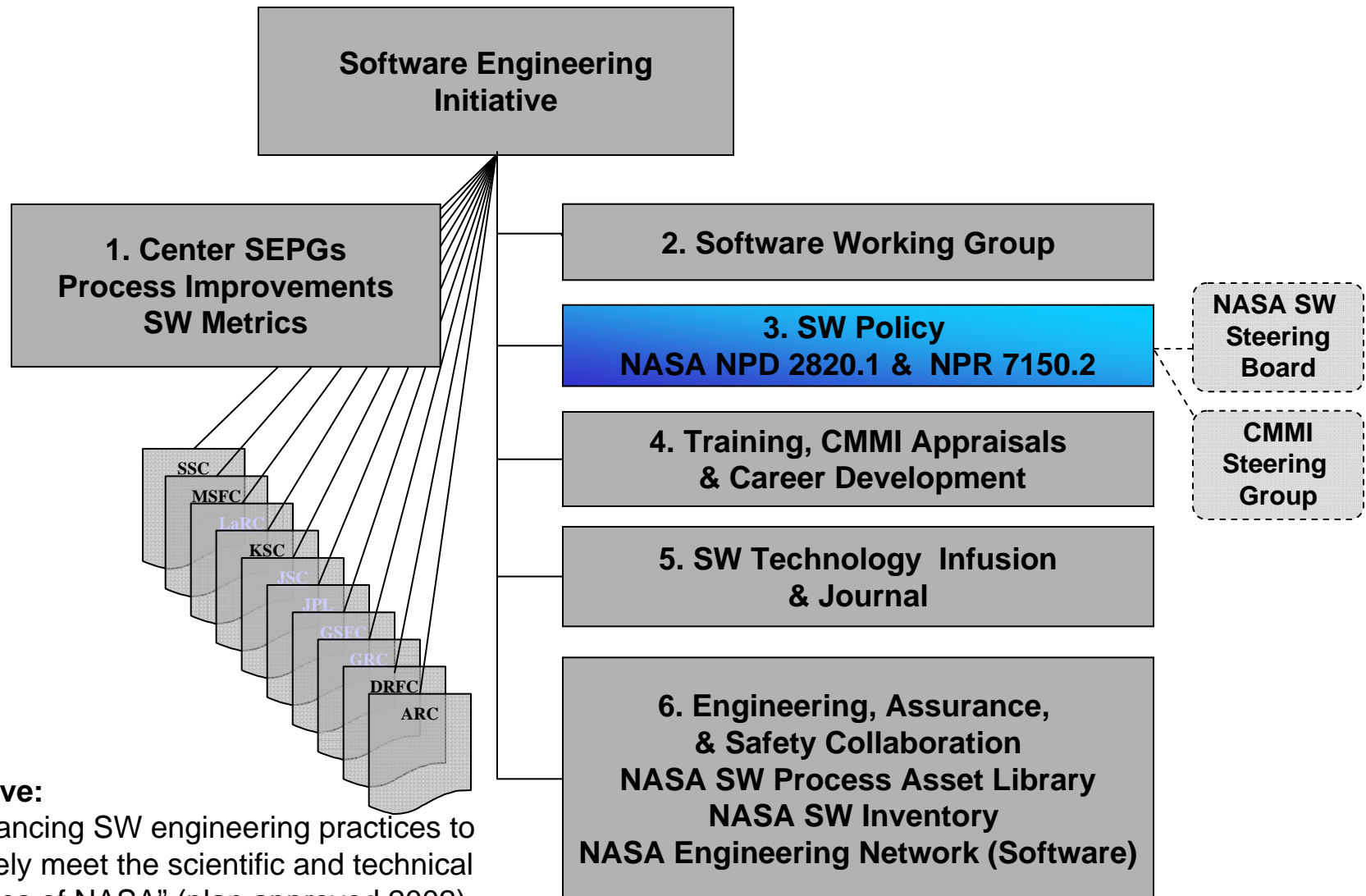


Objectives of NASA Software Working Group Face-to-Face Meetings

- Facilitate communication across NASA and Software Engineering Institute with respect to providing lessons learned, process improvement and experiences
- Facilitate communication across all NASA Centers in respect to providing lessons learned on implementing process improvement
- Educate the NASA workforce on best practices in Software Engineering
- Improve the working relationships between Engineering and Safety and Mission Assurance in respect to software engineering
- Improve NASA ability to manage complex software projects
- **Reduce the risk on future NASA missions**



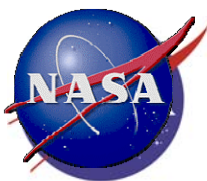
Software Engineering Initiative



Objective:

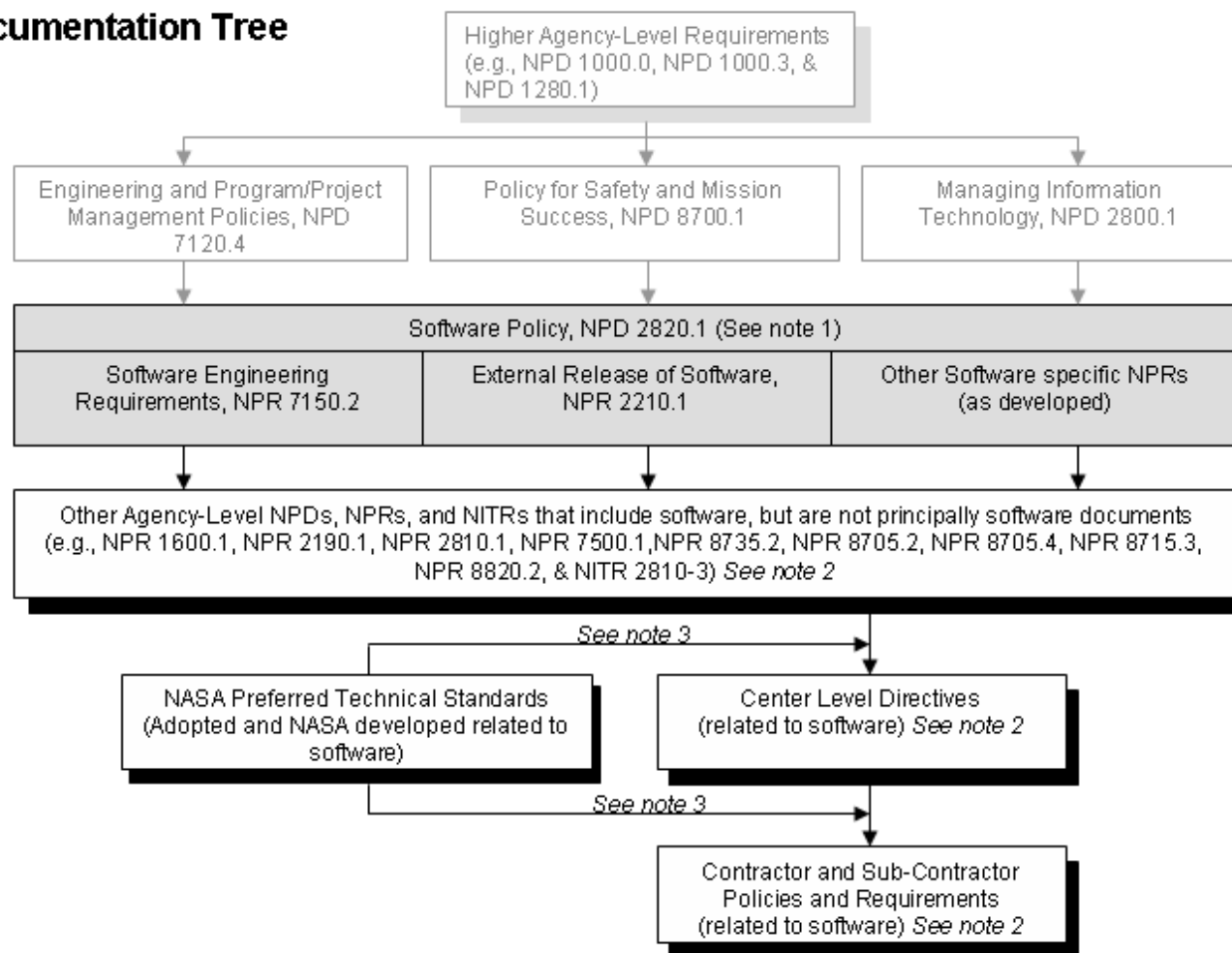
"...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer



NASA SW Documentation Tree (NPD 2820.1)

Attachment 3: NASA Software Documentation Tree



Note 1: NPD 2820.1 is the unique parent document for the Agency's current and future NPRs that are software specific.

Note 2: Other Agency-Level, Center-Level, and Contractor/Sub-Contractor directives can elaborate or augment software specific NPDs and NPRs, but they cannot supersede nor waive NPD 2820.1, NPR 7150.2, or NPR 2210.1.

Note 3: Individual NASA Preferred Technical Standards are required when invoked by an NPD, NPR, Center-Level Directive, contract clause, specification, statement of work, or NASA Technical Warrant Holder. Note that the set of NASA Preferred Technical Standards also includes non-invoked software standards which provide valuable guidance and best practices.



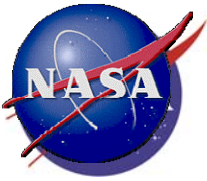
Software is not all the same

flight software ↔ *non-flight software*

engineering software ↔ *general purpose software*

safety critical software ↔ *non-safety critical software*

... and it shouldn't be treated the same!



NASA-wide Software Classification*

OCE	Class A	Space Flight Human Rated Software Systems
	Class B	Non-Human Space Rated Software Systems
	Class C	Mission Support Software & Facilities
	Class D	Analysis and Distribution Software
	Class E	Development Support Software
CIO	Class F	General Purpose Computing Software (Multi-Center or Multi-Program/Project)
	Class G	General Purpose Computing Software (Single Center or Project)
	Class H	General Purpose Desktop Software

- Notes** 1. "It is not uncommon for a project to contain multiple systems and subsystems having different software classes" (P.2.1)
2. Whether software is safety critical is an independent determination based on NASA-STD 8719.13

* Established by NPR 7150.2

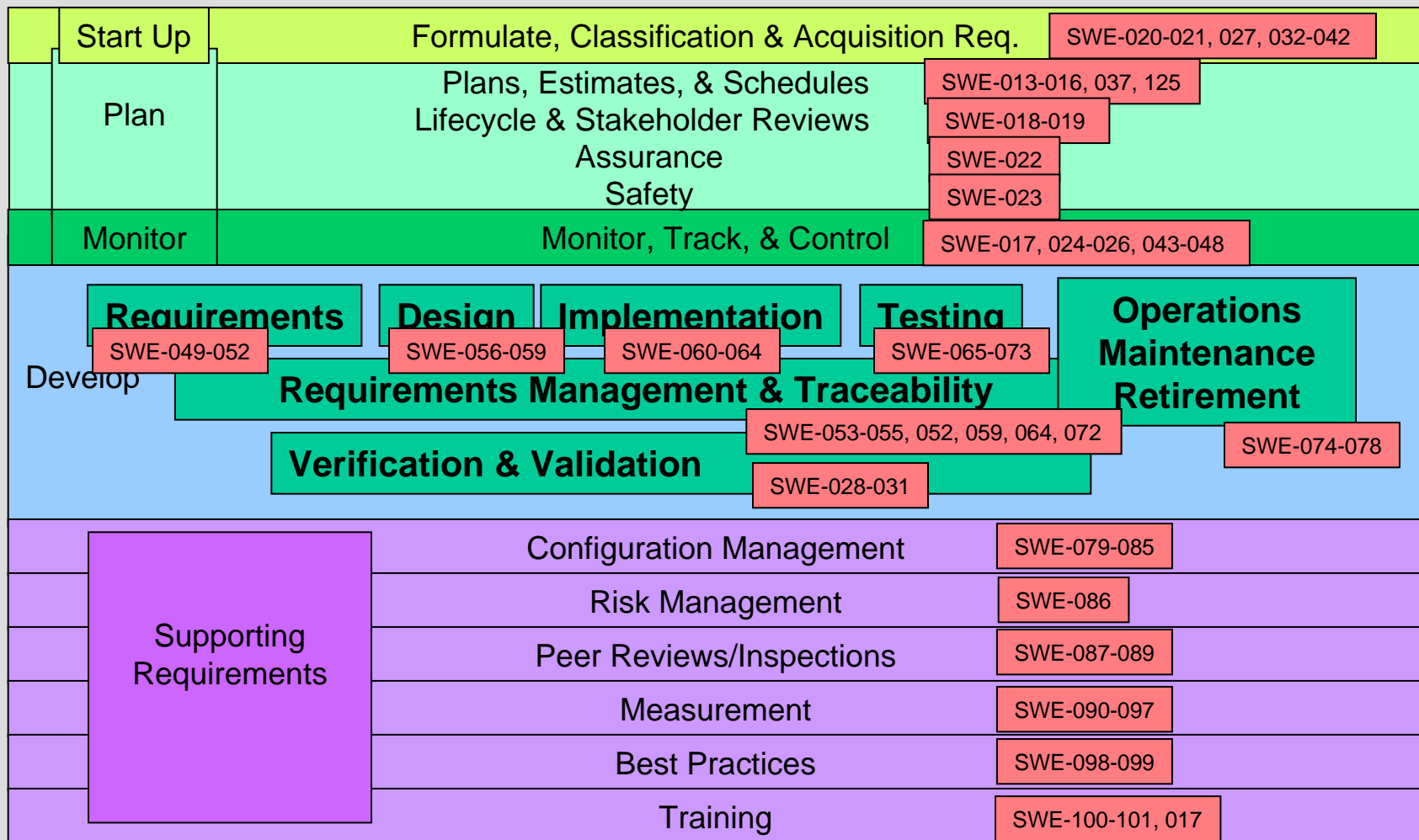


NPR 7150.2 SW Engineering Requirements in a Nutshell (for projects, excluding specific documentation content)

Relevant SW Laws, Policies and Requirements:

SWE-007-012

Disclosure, Technology Transfer, External Release, Security, Disabilities





Applicable and Non-Applicable Software Engineering Requirements (NPR 7150.2)

← OCE Classes → ← CIO Classes →

				Class A	Class B	Class C	Class D	Class E	Class F	Class G	Class H
SW Requirements Development	<i>Document</i>	49	Project	X	X	X	X	P (Center)	X	X	
	<i>SW requirements</i>	50	Project	X	X	X	X		X	X	
	<i>Flow-down & derived req.</i>	51	Project	X	X	X			X (not OTS)	P (Center)	
	<i>Bi-directional trace</i>	52	Project	X	X				X (not OTS)	P (Center)	
SW Requirements Management	<i>Manage req. change</i>	53	Project	X	X	X	X	P (Center)	X	X	
	<i>Corrective action</i>	54	Project	X	X				X	X	
	<i>Requirements Validation</i>	55	Project	X	X	X			X	X	
SW Design	<i>Document design</i>	56	Project	X	X	P (Center)			X (not OTS)	X (not OTS)	
	<i>Architecture</i>	57	Project	X	X	P (Center)	P (Center)		X (not OTS)	X (not OTS)	
	<i>Detailed design</i>	58	Project	X	X				X (not OTS)	X (not OTS)	
	<i>Bi-directional trace</i>	59	Project	X	X				X (not OTS)	X (not OTS)	
SW Implementation	<i>Design --> code</i>	60	Project	X	X	X	X		X (not OTS)	X (not OTS)	
	<i>Coding standards</i>	61	Project	X	X				X (not OTS)	X (not OTS)	
	<i>Unit test</i>	62	Project	X	X	X	P (Center)	P (Center)	X (not OTS)	X (not OTS)	
	<i>Version Description</i>	63	Project	X	X	P (Center)	X		X (not OTS)	X (not OTS)	
	<i>Maintain Traceability</i>	64	Project	X	X				X (not OTS)	X (not OTS)	
SW Testing	<i>Plan, procedures, reports</i>	65	Project	X	X	X	P (Center)		X	P (Center)	
	<i>Perform testing</i>	66	Project	X	X	X	X		X	P (Center)	
	<i>Test for compliance</i>	67	Project	X	X	X			X	P (Center)	
	<i>Evaluate test results</i>	68	Project	X	X	X	X		X	P (Center)	
	<i>Doc. defect & track</i>	69	Project	X	X	X	P (Center)		X	P (Center)	
	<i>Models, simulations, tools</i>	70	Project	X	X				X	P (Center)	
	<i>Update plans & procedures</i>	71	Project	X	X	X			X	P (Center)	
	<i>Maintain Traceability</i>	72	Project	X	X	X	X		X	P (Center)	
	<i>Platform or Hi-Fidelity sim.</i>	73	Project	X	X	X			X	P (Center)	
	<i>SW Data Dictionary</i>	110	Project	X	P (Center)				X (not OTS)	X (not OTS)	

X = Required
Blank = Not Required
P(Center) = Center defined process (i.e. subset OK)

“Through the use of the Requirements Mapping Matrix, the number of applicable requirements and their associated rigor are scaled back for less critical software classes”,
- NPR 7150.2, Section P.2.1

**Waiver Authority for NPR
7150.2, Software Engineering
Requirements**

*Table indicates
which level of
Engineering
Technical Authority
is needed for a
waiver*

*89% of the project implemented
requirements are under Center
level waiver authority*

Section of NPR	Requirement Descriptor*	SWE Requirement #	Implements Requirement	Waiver Authority for NPR 7150.2 by Req
SW Life Cycle Planning	<i>SW Plan</i>	13	Project	Center Director*
	<i>Execute Plan</i>	14	Project	Center Director*
	<i>Cost Estimation</i>	15	Project	Center Director*
	<i>Schedule</i>	16	Project	Center Director*
	<i>Training</i>	17	Project	Center Director*
	<i>Reviews</i>	18	Project	Center Director*
	<i>Life Cycle</i>	19	Project	Center Director*
	<i>SW Classification</i>	20	Project	HQ OCE
	<i>SW Classification changes</i>	21	Project	Center Director*
	<i>SW Assurance</i>	22	Project	HQ OCE/OSMA
	<i>SW Safety</i>	23	Project	HQ OCE/OSMA
	<i>Plan Tracking</i>	24	Project	Center Director*
	<i>Corrective Action</i>	25	Project	Center Director*
	<i>Changes</i>	26	Project	Center Director*
Off The Shelf (OTS) SW	<i>COTS, GOTS, MOTS</i>	27	Project	Center Director*
Verification & Validation	<i>Verification planning</i>	28	Project	Center Director*
	<i>Validation planning</i>	29	Project	Center Director*
	<i>Verification results</i>	30	Project	Center Director*
	<i>Validation results</i>	31	Project	Center Director*

** Center Director or designee (e.g. Center Engineering Director or Lead Discipline Engineer)*



Constellation Program Computing Systems Requirements CxP 70065

Revision: Draft Baseline	Document No: CxP 70065
Release Date: TBD	Page: 6 of 101
Title: Constellation Program Computing Systems Requirements	

**Example of the
NASA Software
Initiative
making an impact
on Critical NASA
Programs**

98 2.0 DOCUMENTS

99 2.1 APPLICABLE DOCUMENTS

100 The following documents include specifications, models, standards, guidelines,
101 handbooks, and other special publications. The documents listed in this
102 paragraph are applicable to the extent specified herein.
103

NPR-7150.2 NASA Software Engineering Requirements

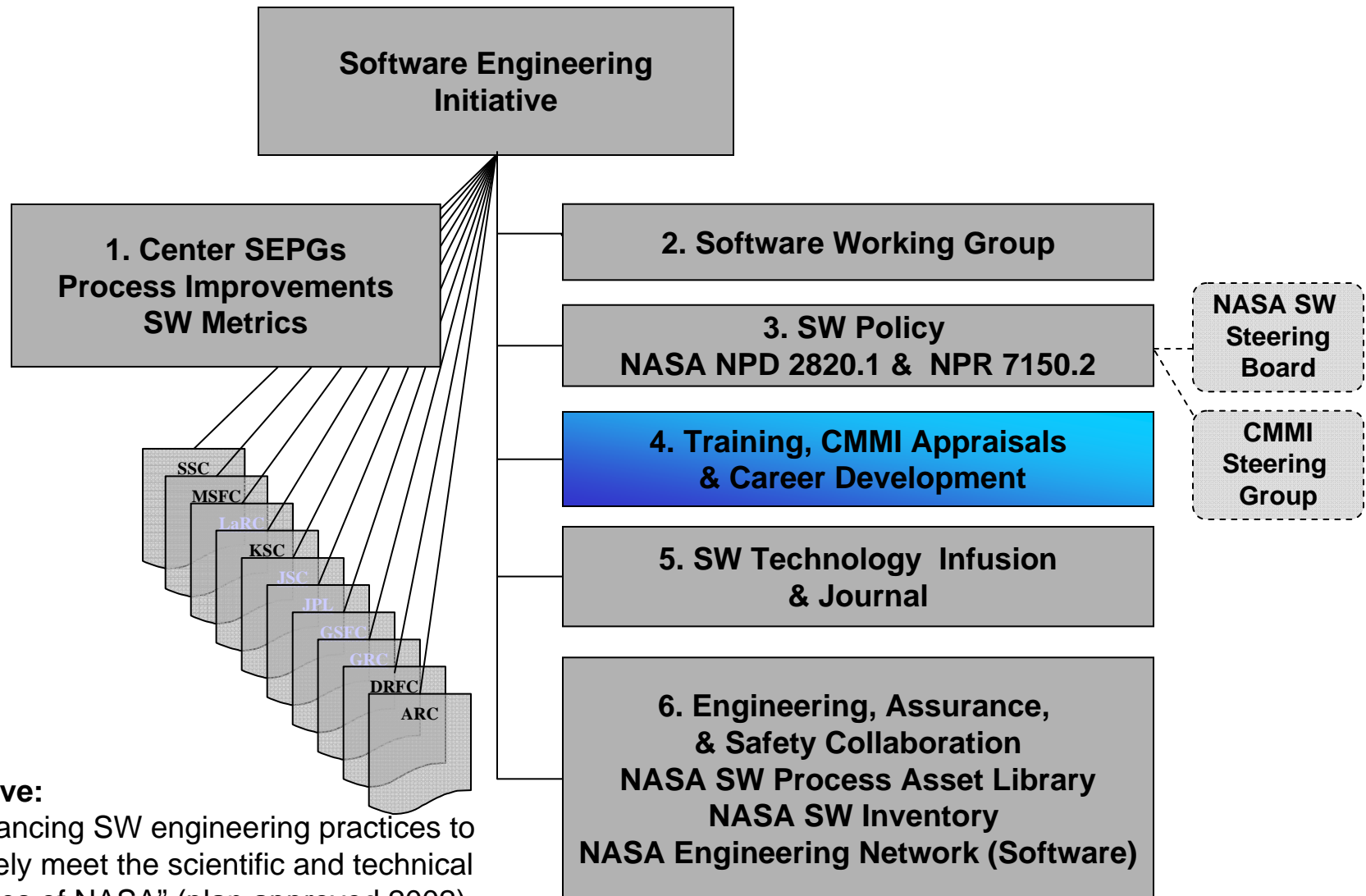
140 **Software engineering.** Each project shall adhere to the requirements of NPR
141 7150.2 Software Engineering Requirements. [CSR-31-003]

142 *Rationale: NPR 7150.2 is an agency-level requirements document applicable to*
143 *all projects per NASA Policy Directive (NPD) 2820.1 NASA Software Policies.*
144 *The Constellation Program System Engineering Management Plan (CxP-70013)*
145 *describes the independent technical authority role for the program. Note that the*
146 *independent technical authority must approve all variances to the Software*
147 *Engineering Requirements per SWE-102 and NPR 7150.2 chapter 6.*

744 **Safety plan.** Each project shall prepare a Software Safety Plan per Section 5.2
745 of NASA-STD-8719.13B Software Safety. [CSR-34-011]



Software Engineering Initiative



Objective:

"...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer



March 14th Baseline
Version 5.5

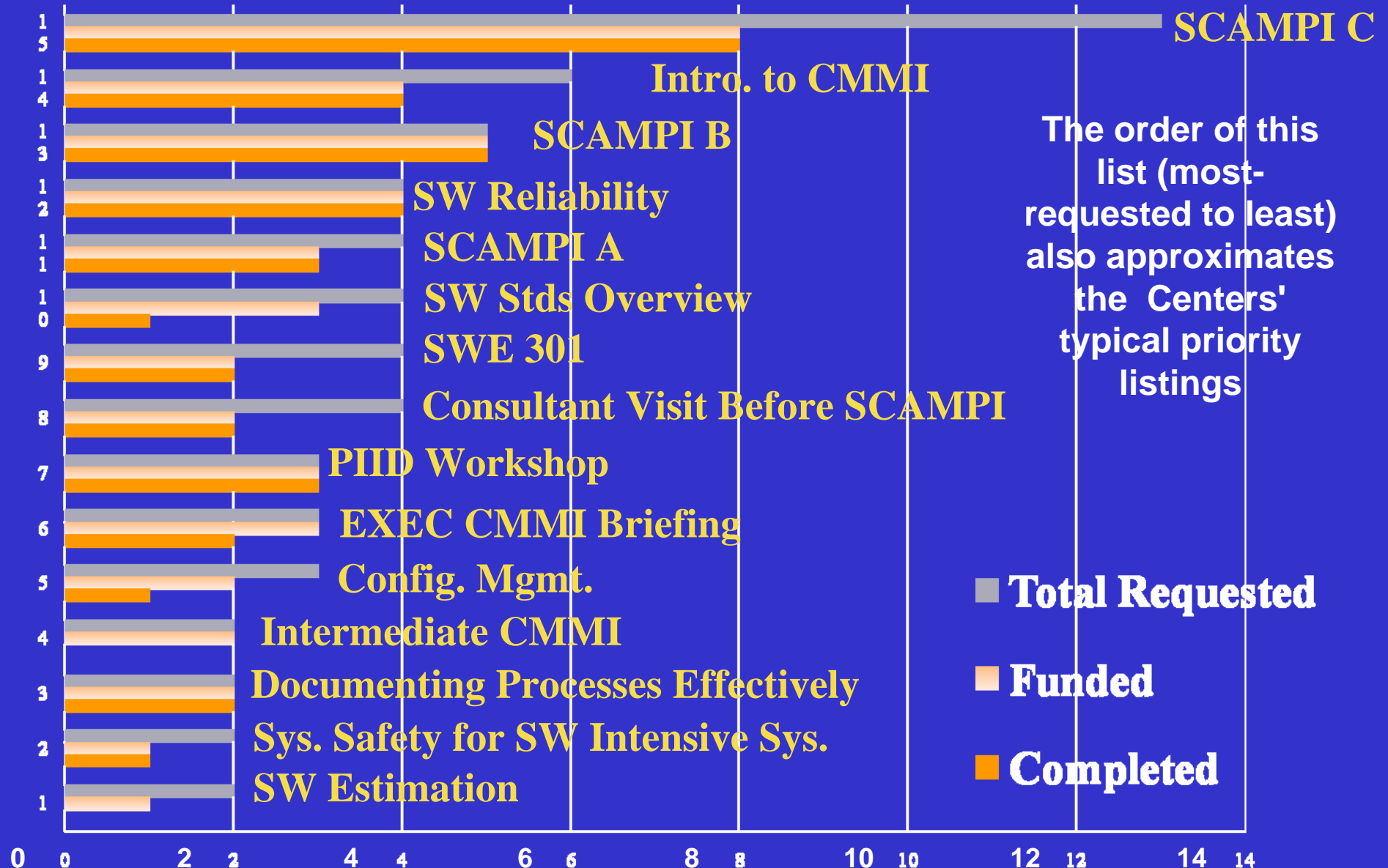
RUNNING STATUS ON SOFTWARE ENGINEERING DACUM CURRICULUM

11. Software Engineering 301
12. Software and Process Metrics
13. Software Estimations
14. Software Process Improvement
15. Software Acquisition
16. Formal Methods for Software

Structured On-the-Job Learning

Informal On-the-Job Learning

FY 08 Training Requests



Of the 20 additional single courses requested once each, 4 were completed, 8 are of interest to both SWE/SWA, and the 20 cover 38 unique DACUM skills.

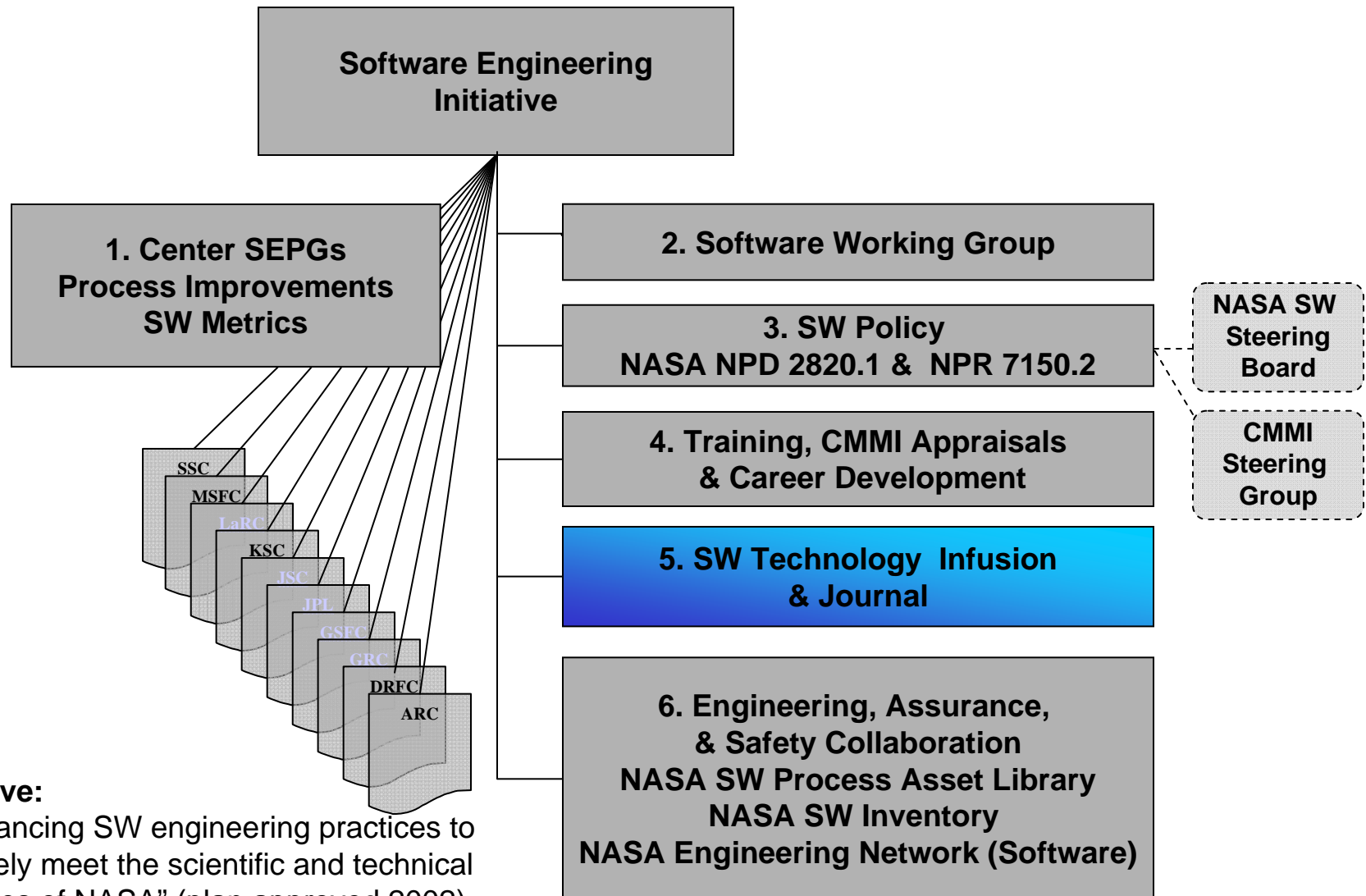


SWE 301 Objectives

- The objectives of the class are as follows:
 - Have participants *understand the 41 defined Requirements* for Software Engineering Management
 - Have participants *focus on the application of project management techniques* as applied to software projects
- To meet these objectives there are 9 presentation modules and 8 workshop modules
- Has been offered twice (East coast and West Coast), offering again in FY09



Software Engineering Initiative



Objective:

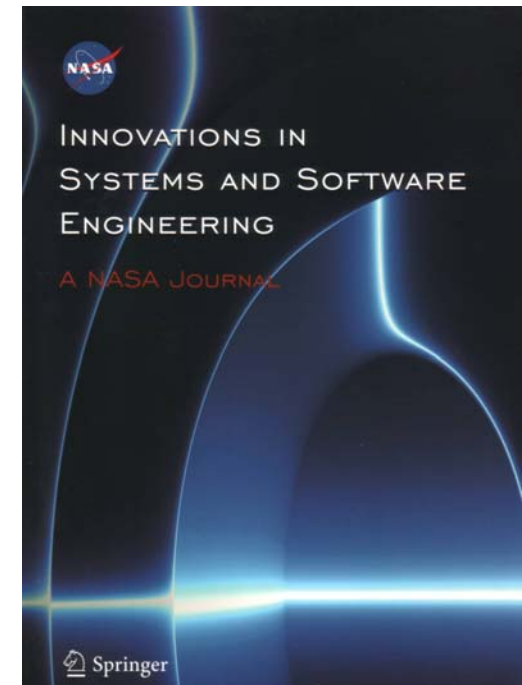
"...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

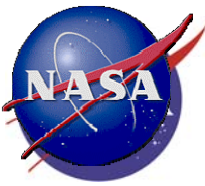
Sponsor: NASA Office of Chief Engineer



Research & Technology Infusion Journal

- Encourage and stimulate advanced technology and research work that is relevant to NASA's missions
- Promote positive communication between the research and practicing professionals
- Supports the maturation of software and systems engineering related to aerospace applications
- Joint editorship between academic and government researchers
- Kickoffed in April 2005





Research Infusion Collaborations (Joint OCE & OSMA task)

- **2005**

- Application of Software Cost Reduction Methods and Tools to ISS Science Payload Software
- Application of Design Advisor to Instrument Flight Software
- Infuse Code Surfer into NASA IV&V Process
- Application of SpecTRM for Design Rationale Capture and Hazard Analysis during Conceptual, Concurrent Design of Missions

- **2004**

- Can CodeSurfer Increase Inspection Efficiency?
- Analysis of MSFC Flight Software
- Finding Defect Patterns in Reused Code
- ISS Software Analysis
- Cross-Project Application of Perspective-Based Inspection
- Perspective-Based Inspection Infusion

- **2006:**

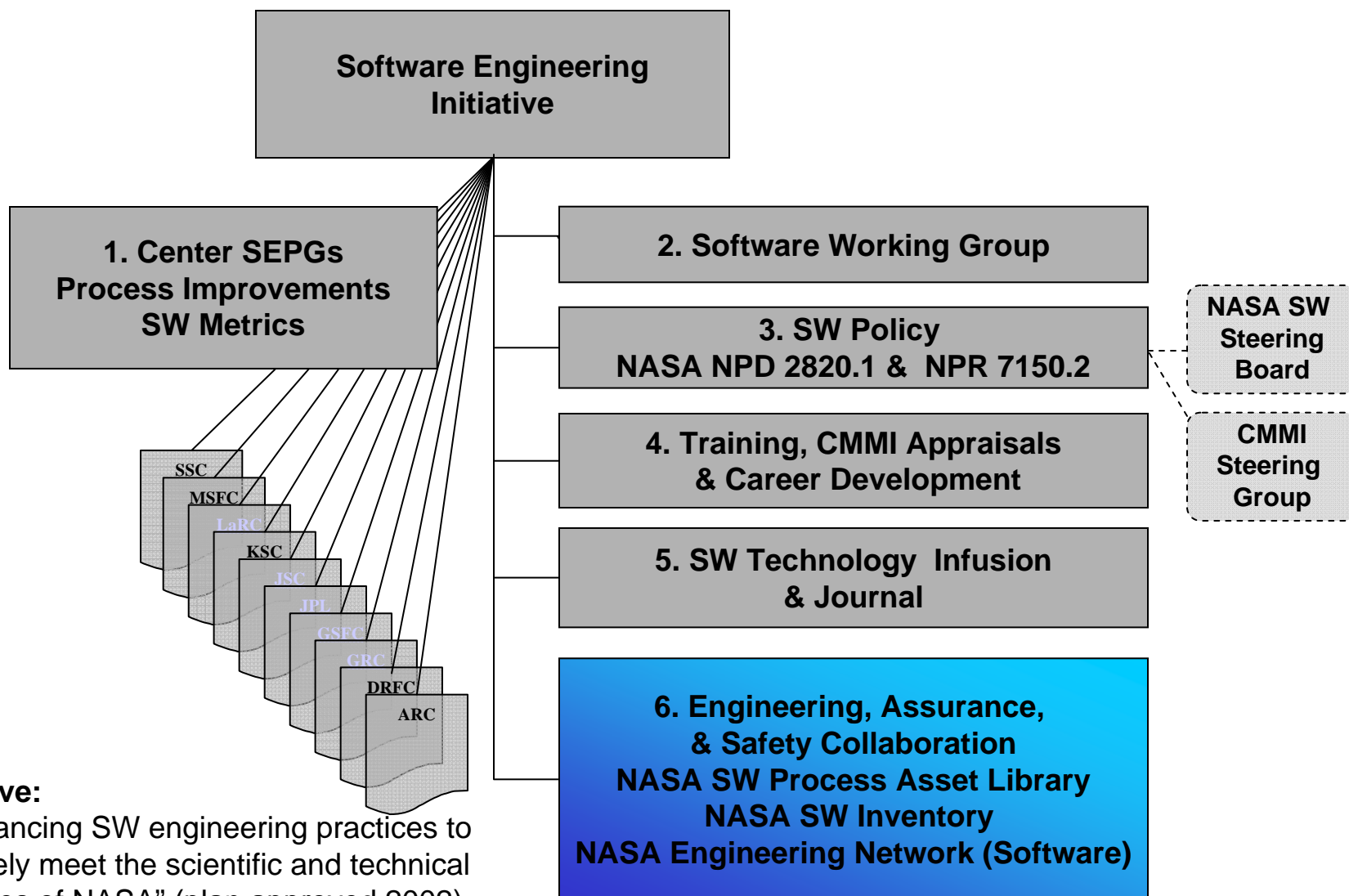
- Infusion of CodeSurfer into TCMS Sustaining
- Technology Infusion of Klocwork InSpect into Flight Software Branch Development Processes at GSFC
- Technology Infusion of Source Code Analysis and Error Checking into Mission Planning and Execution Software Tools at Jet Propulsion Laboratory
- Technology Infusion of Software Architecture Visualization and Evaluation (SAVE) into the Common Ground Software Development Process for NASA Missions at JHU/APL

- **2007**

- Infusion of Perspective Based Inspections at NASA IV&V
- Support Model-Based Systems and SW Engineering with SpecTRM (JPL)
- Tech. Infusion of CodeSonar into the Space Network Ground Segment at GSFC/White Sands
- Technology Infusion of SDA into MOD SW Development Process (JSC)
- Technology Infusion of SAVE into STRS Architecture Compliance Verification at GRC
- Infusion of Requirement Assistance into CEV IV&V Validation Activities



Software Engineering Initiative



Objective:

"...advancing SW engineering practices to effectively meet the scientific and technical objectives of NASA" (plan approved 2002)

Sponsor: NASA Office of Chief Engineer



NASA Engineering Network Software Engineering Portal


Welcome to Inside NASA Site Map | Login | Register

Home Business Centers Education Employees Engineers Emergency Operations Managers News & Library Help & Feedback

Nondestructive Evaluation Knowledge Management **Software Engineering** Structures Systems Engineering

Software Engineering

Welcome (SW)



Welcome to the Software Engineering Community. I am the Leader of the Software Engineering Community of Practice. I welcome your ideas and experiences. --John Kelly

[+ Read Biography](#)

Contact: [John C. Kelly](#)

Co-Facilitators: [Keri Murphy](#), [Erin Means](#)

Center Sites (Software)

- Ames Software Process Improvement
- Glenn Software Engineering Process Group
- Goddard Software Process Improvement
- IV&V Facility
- Jet Propulsion Lab Software Site (JPL Only)
- Johnson Software Site
- Langley Software Process Improvement Initiative

Agency Links (Software)

- [NESC \(Software\)](#)
- [Software Assurance](#)
- [software.nasa.gov](#)
- [Software Process Asset Library \(PAL\)](#)
- [Software of the Year Award](#)

NASA Technology and Research (Software)

- [ARC Robust Software Engineering Group](#)
- [ARC Software Engineering Research Infusion](#)
- [Innovations in Software and Systems Engineering Journal](#)
- [JPL's Laboratory for Reliable Software](#)
- [LaRC Formal Methods Site](#)
- [NASA Software of the Year Award](#)

Announcements--Software

Current New Mine

[NASA Software Working Group Face to Face May 13](#) Apr 27, 2007

Hosted by Stennis Space Center

[« Prev](#) | 1-1 | [Next »](#)

Discussion Boards (SW)

Forum (1-2 of 2)	Msgs	Last Post
Software Metrics	3	1/29/2007 10:03 AM
General Software Topics	5	3/14/2007 10:57 AM

All Forums

Software Process Asset Library (Recent Documents)

- [- Software Project Schedule Development](#) - 228KB
Initial COCOMObased schedule and resource estimates, List of required software project reviews, Master project schedule, Metrics from similar past projects 2.2 Adjust the COCOMO-based schedule and...
[+ Read More](#)
Creator: cfarrell
Collection: SWPAL
- [- 12.0 Testing Phase](#) - 244KB
Figure 12-2 System Test Process Flow System Test Support Entrance Criteria This lifecycle phase follows formal software test or may occur in parallel with formal software test. Using the STP and...
[+ Read More](#)
Creator: Padmin
Collection: SWPAL

Find it @ NEN

The NASA Engineering Network (NEN) search provides the capability to search across a variety of NASA Engineering repositories. Enter a search term in the text box below and click GO. Click on the advanced search button to refine your search.

[+ GO](#)


[+ ADVANCED SEARCH](#)

You may also search on the following categories.

- [NASA Centers](#)
- [Mission Directorates](#)
- [Topics](#)
- [By Year](#)
- [By Collection](#)

Point of Contact List (Software)

[Click here](#) for a list of Points of Contact and biographical information within the Software Engineering Community.



Suggestions (Software)

Let us know what you think! Click on the button below to make a suggestion.



NASA Process Asset Library

Category — PAL - Windows Internet Explorer

http://swpal.msfc.nasa.gov/category

File Edit View Favorites Tools Help

Category — PAL

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Software Process Asset Library (PAL)

Home Life-Cycle Category Agency Center CMMI Feedback

You are here: Home » Category

You are not logged in : Log in

Navigation

- Home
- Life-Cycle
- Category
 - Requirements
 - Checklists
 - Compliance Matrices
 - Software Surveys
 - Handbooks and Guides
 - Development and Coding Guidelines
 - Studies and Reports
 - Software Project Management
 - Software Cost Estimation
 - Project-Defined Processes
 - NASA Center Specific Processes
 - Domain-Specific Processes
 - Documents (Reference & Project-Specific)
 - Software Engineering Models
 - Software Measurements
 - Templates
 - Presentations & Procedures

Category

Up one level

- Requirements
- Checklists
- Compliance Matrices
- Software Surveys
- Handbooks and Guides
- Development and Coding Guidelines
- Studies and Reports
- Software Project Management
- Software Cost Estimation
- Project-Defined Processes
- NASA Center Specific Processes

http://swpal.nasa.gov/

The NASA PAL has two goals:
Encourage sharing of software engineering assets within the agency.
Provide a repository of software engineering needs.

The NASA PAL contains pointers to over 650 assets
Effective communication of templates, guides, checklist, and procedures between Centers
Encourage Centers to continue to share their software related assets across the Agency

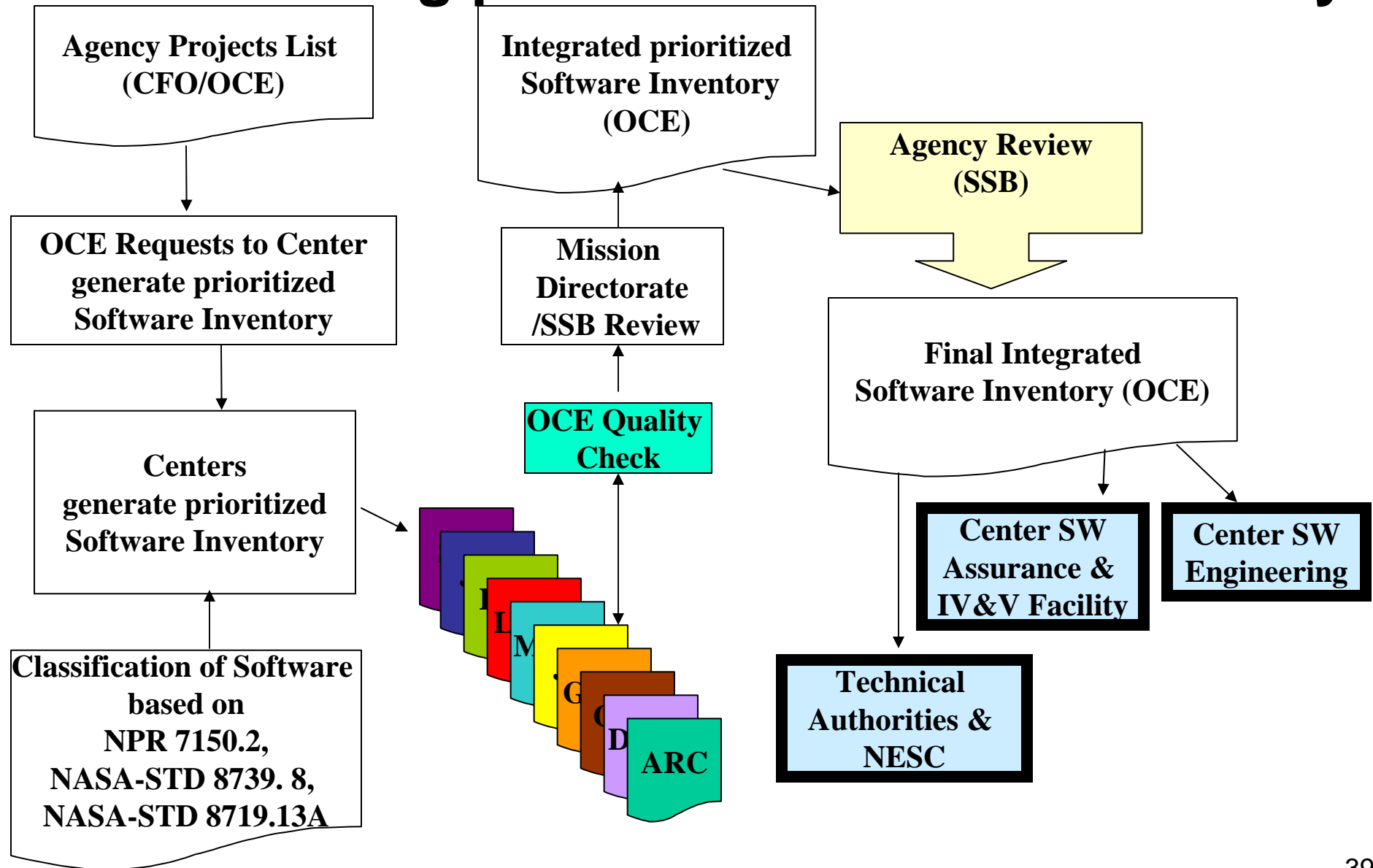
start

Windows E... Inbox - Microsof... Verifying planne... FW: Internal Re... 6 Microsoft Of... FY2008-09 High... V2-5-Draft-715... V3-2009-softwa... Category — PA...

Trusted sites 100% 9:25 AM

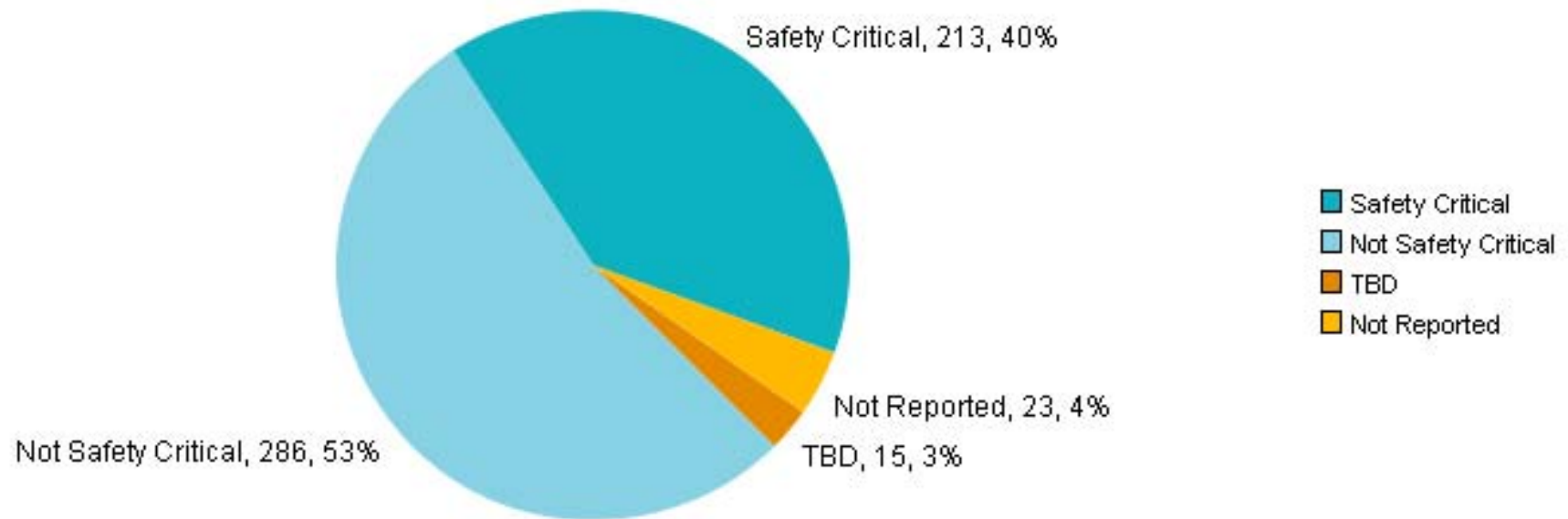


Annual Process for Compiling and Validating prioritized Software Inventory





2008 NASA Safety Critically (Software Class A,B,C Projects)



Notes:

I. Deltas between 2007

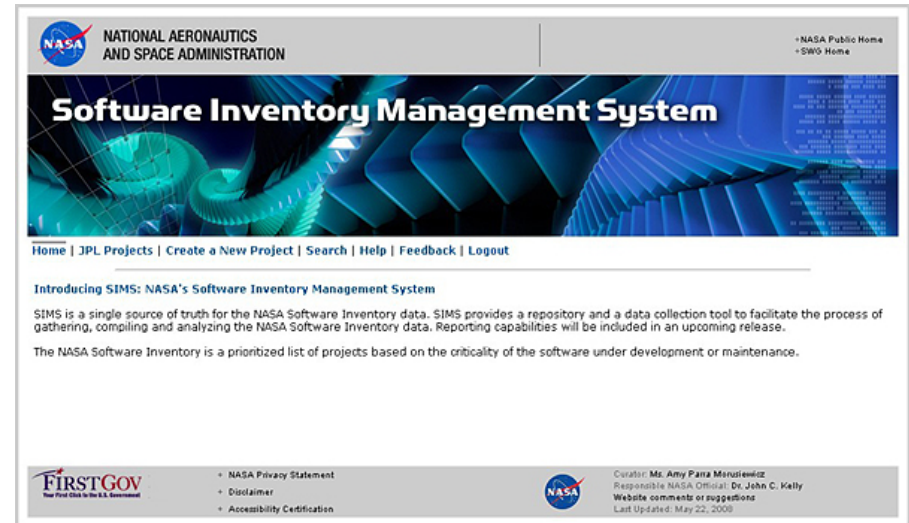
1. Reduced "Not Reported" from 14% to 4%
2. Safety Critical increased from 38% to 40%

Total Center level projects containing
Class A, B, and/or C software = 537



Highlight: Software Inventory Management System

- New tool in 2008
- Serves as a repository & data collection tool to facilitate the process of gathering, compiling and analyzing the NASA Software Inventory data
- JPL developers have been very responsive and have provided reporting capabilities earlier than originally scheduled.
- Over 1000 entries





Software Toolshed

Focus Area/Concept/Objective

- Deploy “in-house”, open source and COTS software tools (static analysis) on NASA mission software.
- Demonstrate the feasibility of an Application Service Provider model that provides missions with software tools and expertise.
- Taking a general approach to ASP; Language independent but a focus on Java with some C/C++ capability.
- Recognizing the need for “bug” reporting strategies for developers and managers.
- Building a relationship between NASA software engineering research, applied software engineering and mission software development (infusion and requirements).



Software Engineering Initiative

- Reduces risk of software failure - Increases mission safety
- More predictable software cost estimates and delivery schedules
- Smarter buyer of contracted out software
- More defects found and removed earlier
- Reduces duplication of efforts between projects
- Increases ability to meet the challenges of evolving software technology